

REMARKS

This application has been reviewed in light of the Office Action dated June 2, 2009.

Claim 12 is pending in the application. By the present amendment, claim 12 has been amended. Support for this amendment may be found at least in Figs. 3-4, page 12 line 20 - page 13 line 3, page 14 lines 10-20, and page 19 lines 6-15 of the present specification. No new matter has been added. The Examiner's reconsideration of the rejection in view of the following arguments/remarks is respectfully requested.

By the Office Action, claim 12 stands as rejected under 35 U.S.C. 103 (a) as being unpatentable over U.S. Patent No. 7,426,738 to Deverill et al. (hereinafter "Deverill") in view of U.S. Patent No. 6,725,243 to Snapp (hereinafter "Snapp") in further view of U.S. Patent No. 6,738,798 to Ploetz et al. (hereinafter "Ploetz"). Applicants respectfully assert that Deverill, Snapp and Ploetz, taken alone or in combination, do not teach or suggest claim 12 of the present invention.

Deverill and Snapp fail to teach or suggest "*marking the earlier measurement as changed and adding the new measurement to the list,*" as recited in claim 12. This point has been acknowledged by the Examiner in the Office Action and during the interview conducted on May 29, 2009. Thus, the applicants will not comment any further on this issue.

Ploetz fails to cure the deficiencies of Deverill and Snapp. As discussed in the May 29 interview, Ploetz discloses that the 'Contracts' and 'Scanner Status' tables contain only one record for each scanner and that these values are changed/modified when the relevant information changes (see e.g., col. 7, line 59-60; col. 10, lines 63-65; and col. 12, lines 38-55). The portions of Ploetz cited by the Examiner only teach keeping a history of the changes made

to the ‘Contracts’ and ‘Scanner Status’ tables in a separate ‘Audit’ table, i.e. Contract Audit Table is history for Contract Table (see col. 7, line 66 – col. 8, line 4); Scanner Status Audit Table is history for Scanner Status Table (see col. 11, lines 51-55). In contrast, the present invention stores the earlier measurements and new measurements in the same measurement list (See Fig. 4 and page 14, lines 14-19). The Examiner suggests that the claims be amended to more distinctly recite this difference.

Although the applicants believe that claim 12, as previously presented, is patentable over Ploetz, by the present amendment, claim 12 has been amended to recite “wherein the measurements are stored in an order-preserving manner in a single list.” As discussed above, Ploetz explicitly teaches that the history of old values is stored in separate table, not in the same single list as the present values. Furthermore, Ploetz is silent as to the manner in which the data is stored. Hence, Ploetz does not remotely contemplate that the data is stored in an order-preserving manner. Therefore, it is clear that Ploetz does not teach or suggest a measurement list “wherein the measurements are stored in an order-preserving manner in a single list,” as recited in claim 12.

In addition, as also discussed in the May 29 interview, the applicants believe that Deverill does not teach the feature of claim 12 reciting “*measuring code as the code is being loaded*.” The Examiner states in the present Office Action that:

‘as the code is being loaded’ is relative to how the system is developed, i.e., some system consider this right when code enters the device and other would interpret this as the code enters the node and is sent to memory of sorts, i.e., cache, and then once the processor is free it would then look at the code.

The applicants disagree with this interpretation of the claim language. Nevertheless, to further prosecution of this case, claim 12 has been amended to recite “*measuring code as the*

code is being loaded into memory and before execution of the code" (see, e.g. page 13, lines 1-3). Deverill, in direct contrast to this language, discloses taking measurements during the execution of a transaction (see, e.g. col. 4, lines 4-18 which teach measuring the processing time of a transaction). The present invention, however, takes measurements as the code is loaded so that the system can be properly attested and verified before executing the transaction (see, e.g., page 9, lines 7-9). Thus, it is clear that Deverill does not teach or suggest "*measuring code as the code is being loaded into memory and before execution of the code*," as recited in claim 12 as amended.

Snapp does not cure the deficiencies of Deverill in this regard. Snapp is directed to a method for accurately updating information in a database and does not remotely suggest measuring any code. As such, Snapp certainly does not contemplate "*measuring code as the code is being loaded into memory and before execution of the code*," as recited in amended claim 12.

Ploetz does not cure the deficiencies of Deverill and Snapp, as Ploetz, too, does not disclose or suggest "*measuring code as the code is being loaded into memory and before execution of the code*." First, it is clear that no executable code is measured in Ploetz. As Ploetz states in col. 4, lines 37-53, Ploetz only collects operation-related data in log files. Collecting operation-related data does not remotely suggest measuring code as the code is being loaded before execution. Moreover, it is well known in the art that data is added to log files after the execution of a transaction. Therefore, Ploetz does not teach or suggest measuring code as it loaded. Hence, Ploetz certainly does not disclose or suggest "*measuring code as the code is being loaded into memory and before execution of the code*," as recited in claim 12.

Thus, for at least the reasons discussed above, the applicants assert that Deverill, Snapp and Ploetz, taken singly or in combination, fail to obviate claim 12 of the present invention. Reconsideration of the rejection is respectfully requested.

Additionally, the applicants assert that the use and combination of the cited references against the present invention is improper. The present invention is directed to a method of providing attestation in a server execution environment. Attestation is a well known term in the art for verifying the identity and integrity of a system or a program running on a system. Attestation is an important tool often used by remote clients in the client-server environment to ensure that the correct application is running on the server and that the server has not been comprised (*e.g.* by viruses). The present invention includes a set of mechanisms to perform integrity measurements on the server or the applications running on the server in both a dynamic and efficient manner. The method of claim 12 recites one such mechanism by measuring parts of the server execution environment to produce a unique fingerprint for each part, aggregating the unique fingerprints, and sending the aggregated value, along with other information, over a network. It is clear that the cited references are directed to fundamentally different inventions than that recited in claim 12.

Deverill is directed to a system and method for monitoring the performance of computer systems. More specifically, Deverill discloses a method of performance monitoring using transaction latency data. Deverill teaches using data inherent to the transaction being measured (*e.g.* a trade reference) as a unique identifier of the transaction. Deverill then teaches measuring the execution time of every task in an identified transaction and using these measurements to

calculate latency data. Deverill does not disclose or remotely suggest providing attestation in a server execution environment. Clearly, measuring a system's performance by timing the execution of transactions has absolutely nothing to do with attesting to the validity and integrity of a system or a program. Thus, Deverill clearly does not teach or suggest the "method for providing attestation in a server execution environment" recited in claim 12.

Snapp is directed to a method for accurately updating information in a database. Snapp teaches an update process where a subject database to be updated checks its entries against a reference database and an update file. If the entries in the subject database are different from the entries in reference database and the entries in the update file, the subject database entries are changed to match the entries in the update file. Snapp makes no reference whatsoever to remote server attestation. Moreover, Snapp does not make any mention of "providing attestation in a server execution environment," as claimed in claim 12 of the present invention. Thus, one skilled in the art would not consult Snapp when providing attestation in a server execution environment.

Likewise, Ploetz is directed to a method for collecting operational data from remote devices. This data is then used to generate centralized reports which compile or summarize the data, and/or start corrective action if the data indicates a problem with the device. Ploetz is completely silent as to remote server attestation. Clearly, the subject matter of Ploetz is totally unrelated to that of the present invention. Thus, since the inventions of Deverill, Snapp and Ploetz are directed to vastly different subject matters than that of the present invention, one skilled in the art would not look to these references when attesting the integrity of systems.

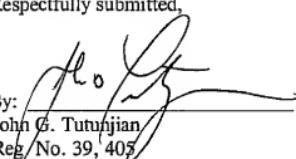
Moreover, as is clear from the descriptions above, the methods of Deverill, Snapp and Ploetz are completely unrelated to each other as well. As such, there is no teaching, suggestion, or motivation to combine the references found either implicitly or explicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art (see, e.g., MPEP §2143.01). Therefore, the applicants disagree with the use of Deverill, Snapp and Ploetz as cited references against the present invention.

In view of the foregoing amendments and remarks, it is respectfully submitted that all the claims now pending in the application are in condition for allowance. Early and favorable reconsideration of the case is respectfully requested.

It is believed that no additional fees or charges are currently due. However, in the event that any additional fees or charges are required at this time in connection with the application, they may be charged to applicant's IBM Deposit Account No. 50-0510.

Respectfully submitted,

Date: July 1, 2004

By: 
John G. Tutunjian
Reg. No. 39, 405

Mailing Address:

KEUSEY, TUTUNJIAN & BITETTO, P.C.
20 Crossways Park North, Suite 210
Woodbury, NY 11797
Tel: (516) 496-3868
Fax: (516) 496-3869